Model-based: End-to-End Molecular Communication System through Deep Reinforcement Learning Auto Encoder

SOHA MOHAMED[1], JIAN DONG[1], A R JUNEJO[2]. DE CHENG ZUO[1]

[1]School of computer science and technology, Harbin Institute of Technology, Harbin 150001, China

[2] School of control science and engineering, Harbin Institute of Technology, Harbin 150001, China

Corresponding author: JIAN DONG(dan@hit.edu.cn).

This research work was supported by the National Natural Science Foundation of China under (No.61100029).

ABSTRACT Molecular Communication (MC) system is an emerging technology for nanoscale networks. Therefore, there is a requirement to develop a new end-to-end MC model which may deliver new perceptions into the aspect of these nanoscale networks. This paper aims to implement the MC framework as an end-to-end Deep Reinforcement Learning (DRL) Auto Encoder (AE). The technique enables training of the MC system without any information about the actual channel (medium) model. For training the receiver and transmitter, the proposed techniques are supervised learning and DRL, respectively. The results show that the performance of the DRL autoencoder (AE) based system achieves nearly the same performance as the traditional modulation and demodulation methods in term of Bit-Error-Rate (BER) under the Gaussian Noise channel but with less complexity. The proposed technique can also be joint with other coding methods to improve their performance.

INDEX TERMS Molecular Communication (MC), Deep Reinforcement Learning (DRL), Auto Encoder (AE), Bit-Error-Rate (BER).

I. INTRODUCTION

Molecular communication (MC) is a bio-inspired technique in which molecules are used for encoding, transferring and receiving information, similar to the way living organs communicate [1]. Due to its bio-compatible, energy-efficient and robust use in biological environments, the MC finds importance in vivo biomedical applications [2], for example, health care monitoring by using the Internet of Bio-Nano Things (IoBNT) [3], empowered bio-Nano sensors, etc.

Currently, the most critical issues in the MC system are how information is reliably transmitted from one end transmitter T_x to another end receiver R_x [4]. During the transmission, the signal suffers from distortion and noise due to the implicated channel state. In the MC, modeling the whole system is optimized in a divide and conquer perspective [5].

Massive research has focused on the optimization of each module for the different channel environments and application demands. According to the data processing theorem [6] in information theory, the optimization of sub-modules for communications cannot assure the global optimal, though it is uncertain if these intelligent modules are adequate to support the greatest probable performance and the efforts to cooperatively optimize these are difficult to be applied [7]. This problem motivates the utilization of Deep Learning (DL) [8] to allow optimization of the MC as an end-to-end system[9-14].In deep learning based communication systems, it is possible to jointly optimize the transmitter and receiver with the structure of an autoencoder [15-18] instead of artificially introduced block schemes[19,20].

DL techniques enable us to design the MC as an end-to-end system because of their ability to estimate any nonlinear function. This kind of scheme, which is built as a full framework learning, precludes the conventional modular arrangement since the scheme is utilized as a single deep neural network and has the promise of accomplishing the optimum end-to-end performance.

This method aims to obtain an input message at each layer of the system. Essentially, this permits a communication system to be adjusted for transmitting information through a different environment. In this paper, we focus on further exploring the application of an autoencoder in the physical layer. Our main contribution is to identify a structure based on a Deep Reinforcement Learning (DRL) Auto Encoder that can jointly optimize the T_x and R_x design which has the next feature, the property of using supervised learning and DRL for training the receiver and the transmitter, respectively. Based on the performance evaluation and analysis, we conclude that the proposed technique achieves the same performance as the traditional molecular communication modeling.

> **Notations:** the lower letters and bold upper letters indicate column vectors and matrices, respectively. The sets of real and complex numbers are described as $\mathbb{R}$ and$\mathbb{C}$. The Jacobian and gradient operator w.r.t and the group of parameters θ are both indicated as ∇_{θ};$(.)^T$ is the transpose operator.

II. RELATED WORK

In previous studies, such as the MC, pheromone signaling, calcium signaling, microtubules, and the bacterium based communication have been proposed. Between these schemes, MC via diffusion is the most effective and energy efficient scheme [21].

An autoencoder end-to-end communication-based system was developed by [22] and (MIMO) has been produced, both of which demonstrate the versatility of this technique [23].

In refs. [24] and [25], the authors described the use of an autoencoder as an end-to-end system. It was proposed to interpret end-to-end communication systems as autoencoders, where T_x, and R_x are implemented as (DNNs). The main drawback of this is theoretically the need for a mathematical channel model to perform the training. This makes its application to real channels of practical interest challenging [26]. The first approach to avoid this problem consists of training the system using a medium model and at the same time, fine-tuning the receiver with measured data. A shortcoming of this approach is the sub-optimal training of the transmitter, which makes it not totally satisfactory [27].

Moreover, other solutions studied by multiple authors consist of learning a differentiable channel computational model in the form of a (GAN), which can then be used for supervised autoencoder training. However, it still needs to be shown that this approach works for practical channels [28]. Lastly, the receiver is not actually trained and detects symbols through clustering. Another model-free approach is developed based on simultaneous perturbation methods. Much research has focused on the optimization [29,30], although all of these methods do not necessitate any knowledge of the channel and can be directly performed with real hardware.

III. MOLECULAR COMMUNICATION SYSTEM MODEL

The essential components of an MC system are a transmitter, receiver, and channel. The transmitter is a point placed at a distance (d) from the midpoint of the receiver. It releases molecules that freely and individually diffuse from each other via Brownian motion. To be precise, the molecules randomly diffuse over the channel, and the gesture of a molecule is not dependent on the gesture of any additional molecules [31].

For each information particle, the hitting ratio is stated as follows:

$$R_{hit}(t) = \frac{r_r}{d+r_r} erfc\left[\frac{d}{\sqrt{4Dt}}\right] \quad (1)$$

Where r, D, and d indicate the radius of the receiver, the diffusion coefficient and the distance between T_x andR_x.

The arrival of molecules is a binomial process in its nature. When considering a single emission of N^{Tx} molecules at t = 0, N_1^{Rx} is expressed as a binomial random variable stated as follows:

$$N_1^{Rx} \sim \mathcal{B}(N^{Tx}, P_1) \qquad (2)$$

Where P_1denotes the predictable amount of molecules absorbed by the receiver node through the initial symbol period, the binomial distribution with n success and the trail probability p is indicated by $\mathcal{B}$ (n; p).

For a broad circumstance with several emissions in a period, the number of received molecules is affected by the existing and prior emissions. Hence, we get

$$N_i^{Rx} \sim \sum_{k=1}^{i} \mathcal{B}(N_k^{Tx}, P_{i-k+1}) \qquad (3)$$

Where N_k^{Tx} indicates the number of emitted molecules in the k^{th}symbol duration.

Due to the complexity of the binomial random variables, the computational model is mostly approached by the Gaussian model [32] given as follows:

$$N_i^{Rx} \sim \aleph\left(\sum_{k=1}^{i} N_k^{Tx}\, P_{i-k+1}\ ,\sum_{k=1}^{i} N_k^{Tx}\, P_{i-k+1}\,(1 - P_{i-k+1})\right) \quad (4)$$

The N_i^{Rx} values are used to evaluate the CDF $F_{N_i^{Rx}}(x)$ for the Gaussian model using the following equation:

$$F_{N_i^{Rx}}(x) = \mathrm{P}(N_i^{Rx} \leq x) \quad (5)$$

Where P(.) refers to the event probability. Fig.1 illustrates the CDF output of Gaussian model.

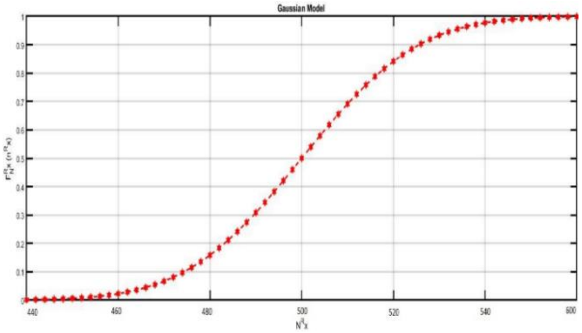

FIGURE 1. The CDF output of Gaussian model

N_i^{Rx} is influenced by the present time and the prior emissions which are being processed by the transmitted bit values with the continuous transmission of bits. In our study, for analysis of the error rate, we use the binary concentration shift keying (BCSK) modulation. In BCSK, bit-0 indicates no emission, and the demodulation is done through simple thresholding. The demodulation function δ (0) receives the input N^{Rx} and the demodulated bits output according to the following equation:

$$Y_i = \delta(\, N_i^{Rx}) \begin{cases} 1 \; if \; N_i^{Rx} \leq \varepsilon \\ 0 \; if \;\; N_i^{Rx} > \varepsilon \end{cases} \quad (6)$$

Where N^{Rx} and Y_i indicate the number of received molecules and the demodulated symbol in the i^{th}symbol period, although ε indicates the threshold value for the demodulation.

The error probability of bit-1 for the Gaussian model, using BCSK for the modulation and thresholding with ε for the demodulation is as follows:

$$P_{e,1}^{G} = \boldsymbol{P}(N^{rx}(t_1, t_1 + t_s) \leq \varepsilon) = \varrho\left(\frac{N^{tx}\ P_{t_1}^{t_1+t_s} - \varepsilon}{\sqrt{N^{tx}\ P_{t_1}^{t_1+t_s}(1 - P_{t_1}^{t_1+t_s})}}\right) \quad (7)$$

Where $\varrho(.)$ refers to the ϱ-function.

A. Noise model

Because of other nano-machines, background molecules and apart from a possible molecular reaction, the channel noise occurs. During demodulation, the gain and calculating consequence molecules could be observed as noises. Gain redundant molecules from other nano-devices called positive noise and the molecules which were collected by other nano-devices are called negative noise [33]. In our study, Additive White Gaussian Noise (AWGN) was used and stated as,

$$N_{noise}\ (\mathrm{n}) \sim \aleph(0, \sigma^2) \quad (8)$$

In MC, the noise proceeds as discrete values, though, for interpretation, its distribution function is estimated as (8). Here, The noise power is described as the variation in the Gaussian distribution.

IV. PROPOSED TECHNIQUE

A. End-to-End DRL Auto Encoder based MC system.

The MC system transmission blocks include a transmitter T_x, receiver R_x and channel as a whole end-to-end DRL over the Gaussian noise model. We now describe in detail all the components of the transceiver as well as the propagation medium. The fully connected end-to-end DRL system is shown in Fig.2.

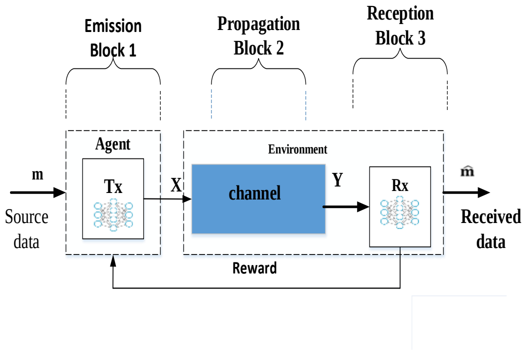

FIGURE 2. DRL Auto Encoder based end-to-end MC system.

An end-to-end MC system is comprised of two nodes whose purpose is to reliably exchange information through the medium. The medium operates as a stochastic model, in which output y and the order probability distribution requirements being met by its input x. The transmitter goal is to send the transmitter T_x information (m) coming from a finite digital set$\{1,\ldots,M\}$, while the goal of the receiver R_x is to identify the referred information from the received signals.

Herein, T_x and R_x implemented as two distinct parametric functions which were cooperatively optimized to meet the application particular performance desires. T_x is stated as $f_{\theta_T}^{(T)}$:M→C^N, where the number of medium uses is N, and the set of parameters is θ_T . The receiver R_x is applied as $f_{\theta_R}^{(R)}: C^N \rightarrow \{P \in R_+^M \mid \sum_{i=1}^{M} P_i = 1\}$ where θ_R is the group of factors and p is a probability vector through the messages. The aim of R_x is to predict m given y by approximating the binomial distribution function. This is completed by learning the conditional log-like hood estimator [34].

$$\theta_R^* = arg_{\theta_R} min\ L(\theta_R) \qquad (9)$$

where L indicates the cross-entropy (CE) shown as

$$L(\theta_R) = \frac{1}{S}\sum_{i=1}^{S} \underbrace{-\log\left(\left[f_{\theta_R}^{(R)}\left(Y^{(i)}\right)\right]_{m^{(i)}}\right)}_{l^{(i)}} \quad (10)$$

Which consider that the training examples are identically and independently distributed (i.i.d.), the size of the training set is S and training set $m^{(i)}$ is the i^{th} training example, the per example loss indicated by $l^{(i)}$ and the received signal is$Y^{(i)}$.

B. Training process

In the Fully Connected (FC) training process, the sequence of training examples $m_{\mathrm{T}}^{(i)}, i = 1,2, \ldots$.. have access to the transmitter and receiver due to pseudorandom number generators set with a similar order. The DRL autoencoder iteration (training techniques) is done through R_x and T_x . This process is to keep working until the stop measure is fulfilled.

The perception of this procedure is that, at every iteration, R_x is enhanced for the fixed θ_T, then the T_x is enhanced for fixed θ_R. The end-to-end MC system should enhance by iteratively doing this procedure. The $f_{\theta_T}^{(T)}$and $f_{\theta_R}^{(R)}$ are differentiable parameters that are adapted over the SGD on the loss function. SGD is the most popular algorithm which is used to do this task [34], which iteratively updates the factors as follows:

SGD (Stochastic gradient descent)

$$\theta^{(j+1)} = \theta^{(j)} - \eta \nabla_\theta \tilde{L}(\theta^{(j)}) \quad (11)$$

The learning rate is $\eta > 0$ and an estimation of the loss function gradient is $\nabla_\theta \tilde{L}$. Through the SGD, the (training dataset) is tested for each iteration to establish a mini-batch, and the gradient of L is estimated with this mini-batch. SGD is applied in both training techniques to adjust the parameters of the T_x and R_x.

C. Training Receiver (R_x)

Training R_x is contacted to the information (m) referred for training. Fig.3 (A) describes the process of the receiver R_x training step by step, and the block model of receiver R_x is shown in Fig.3 (B). The T_x produces a minibatch of size (B_R) of m, which encodes the individual training information into the (N) channel and transfers the mini-batch through the channel. The complex matrix (X) is a (B_R -by-N) matrix which comprise the individual sample of the mini-batch and the related composite information demonstration.

The R_x achieves the new message (y) and produces for the individually training sample a binomial distribution through (M). Lastly, an optimization stage is accomplished by SGD (or a variation) on the (CE) loss function.

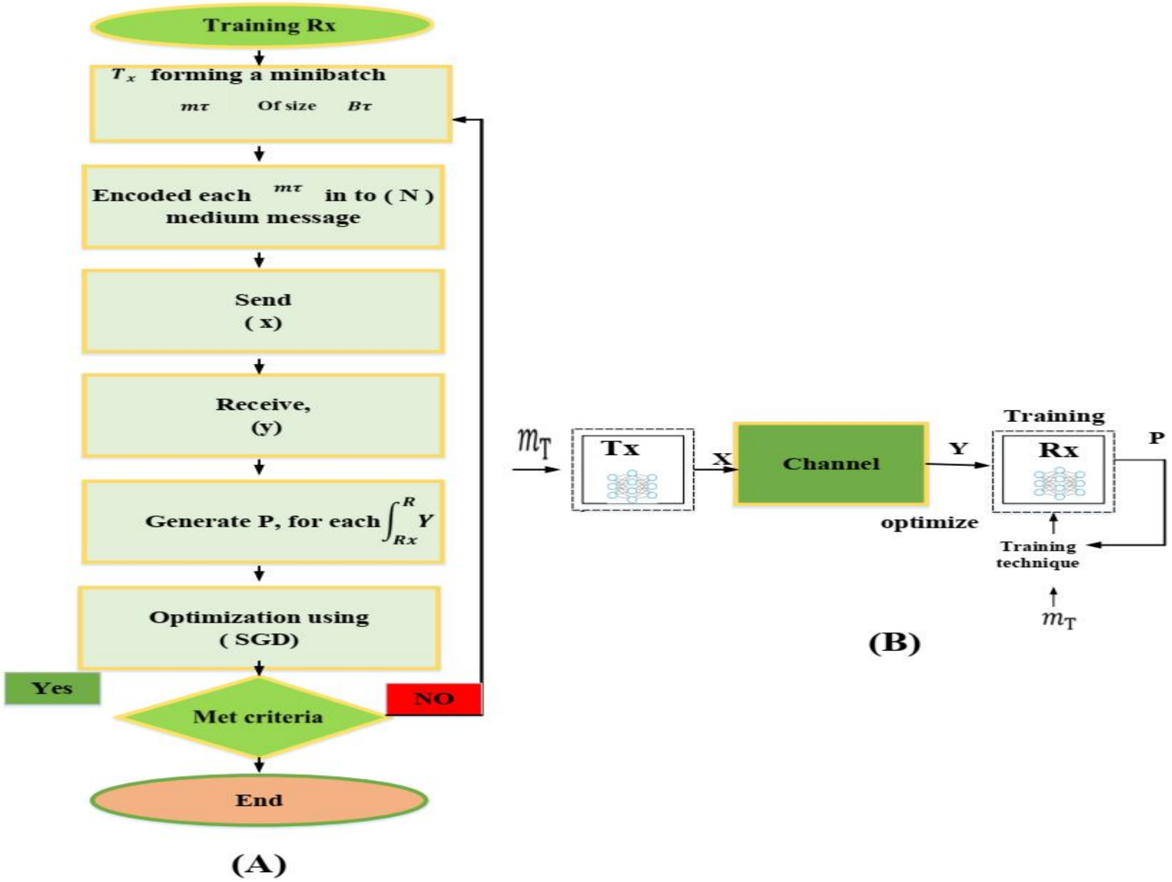

FIGURE 3. (A) Training of the Receiver R_x and (B) block model

D. Training transmitter (T_X)

The transmitter T_x goal is to produce a message that reduces a scalar loss provided by the receiver R_x. This schema relates to a DRL technique. The (M) function is to the state space, and ($\mathbb{C}^N$) is a function of the action space. Moreover, to enable exploration, a random zero-mean perturbation vector (W) is added to the output of the training transmitter (x), and the obtained random vector (X_p) follows the distribution (π_ψ) of the mean($f_{\theta_T}^{(T)}$) which constitutes the stochastic DRL process. The parameter vector (ψ) contains (θ_T) as well as possible additional parameters specific to the distribution of the added perturbation vector, which is only significant at the time of training. The end-to-end performance is indicated by the Loss function (L) and is determined by the dynamics of the channel. (L) is recognized only over the received per example losses l eq (10) delivered by the R_x through an extra trustworthy channel which is required by the procedure of the training. Training samples establishing a mini-batch (m_T)of size (B_T) are first encoded into the channel message (X) to which a perturbation (W) is added to make the perturbed channel message (X_p). The perturbed channel messages are directed through the channel, the receiver achieves the new perturbed (Y) and produces for each training example a probability distribution through (M). The per example losses (L∈ $\mathbb{R}^{B_T}$) are then calculated based on these distributions and the directed messages (m_T). Losses are then directed to the T_x over a trustworthy channel, existing only in the training process. Lastly, SGD or a variant [35] is used to perform an optimization step, where the loss gradient is approximated by

$$\nabla_\psi \widetilde{J}\,(m_T, l, X_p)= \frac{1}{B_T}\sum_{i=1}^{B_T} l^{(i)}\nabla_\psi \log\left(\pi_\psi\left({X_p}^{(i)}\middle| ({m_T}^{(i)})\right)\right) \qquad (12)$$

The training process stops when no more progress is detected. Fig.4. (A) shows the flow chart of the T_x training and Fig.4 (B) shows the DRL model of the T_x training process.

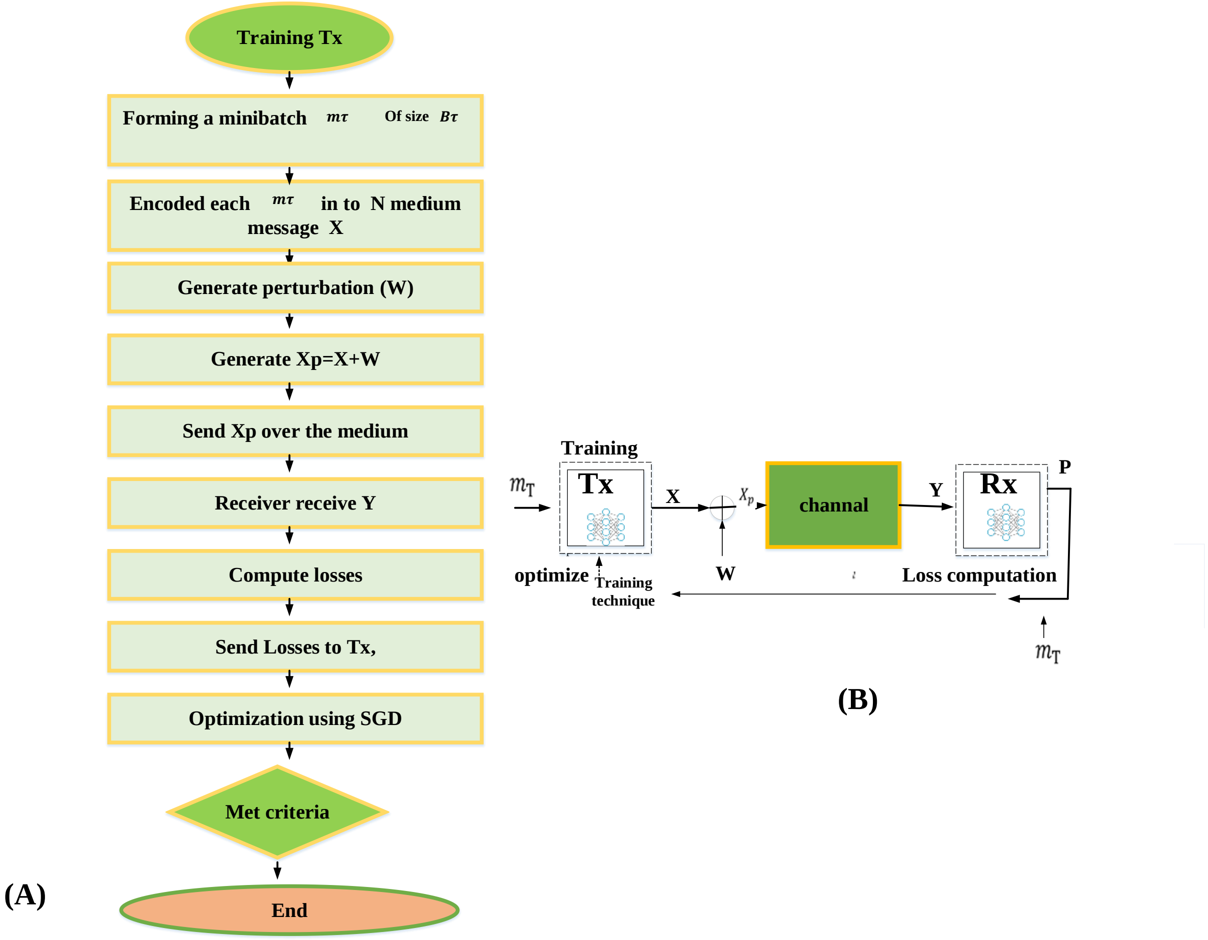

FIGURE 4. DRL (A) Training process and (B) block model of the transmitter T_x .

E. Transmitter T_x and receiver R_x model system.

In our training technique, we perform $f_{\theta_T}^{(T)}$ and $f_{\theta_R}^{(R)}$, as DNNs. A feedforward DNN of K^{th} layers is a parametric function $f_\theta : \mathbb{R}^{N_0} \rightarrow \mathbb{R}^{N_K}$, which maps an input vector $r_0 \in \mathbb{R}^{N_0}$ to an output vector $r_K \in \mathbb{R}^{N_K}$ through succeeding K layers. Every layer calculates an intermediary result or activation vector.

$$r_k = f_{\theta k,K(r_{k-1})} , \qquad k = 1, \dots., K \qquad (13)$$

Where $f_{\theta_k}, k \rightarrow \mathbb{R}^{N_K}$ is the computation assembled by the K^{th} layer, and θ_k is the group of parameters for this layer.

The combination of all the layers is the set of parameters of the all the DNN parameters $\theta = \{\theta_1, \dots\dots, \theta_K\}$. The model of T_x and R_x can take several forms. However, in the context of MC, compound baseband messages are transferred through the MC medium (channel) and the T_x must guarantee the control limitations. The last layer executes normalization, which guarantees that the average energy per message is one. The penultimate layer T_x changes the real outputs of the prior layer into an N - dimensional compound-valued vector, and the T_x model is clearly presented in Fig. 5 a.

The receiver recreates the message directed by the T_x from the received signal, and $f_{\theta_R}^{(R)}$ makes a soft detection by producing a distribution probability p through $\mathcal{M}$, thereafter, hard decoding is completed through selecting the message with a maximum probability.

The R_x initial layer changes the received N compound-valued vector y into 2N real scalars, which are given to a sequence of layers that can be randomly selected.

The $f_{\theta_R}^{(R)}$ latest layer is a soft max layer that performs a probability distribution P over M. Lastly, the message with the maximum probability is selected as a rebuilding of the directed message as in Fig.5b.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2916701, IEEE Access

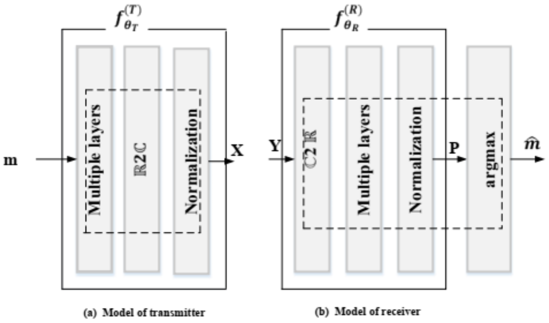

FIGURE 5. Models of transmitter T_x and receiver R_x.

F. Deep Reinforcement learning basics

An agent in a state is shown by s∈ $\mathbb{S}$, which proceeds the action a ∈ A due to policy π [36] . When compelling the action, the agent gets a per example loss l. DRL is used to enhance the mechanism of the agents that interconnect with an environment by taking actions to reduce the loss.

The predictable per-example loss given by a state and action is indicated by $L(s,a)$, i.e, $L(s,a)$=$\mathbb{E}[l \vdots s,a]$. L is presumed to be hidden, and the role of the agent is to discover a policy that reduces the per-example loss.

Through the agent's DRL autoencoder training, the policy π is classically selected to be stochastic, i.e., $\pi(.|s)$ is a probability distribution above the action space A uncertain of the state s. Using a stochastic policy, which allows an investigation of the agent's location, is essential in the DRL. Certainly, training in the DRL is like a try-and-fail procedure; the agent takes an action selected like its state, then recovers its policy as like to the loss acquired from the location. Using a stochastic policy, the agent goal is to reduce the loss $J(s,\pi)$ defined as

$$J(s,\pi) = \int_{a\in A}^{\cdot} \pi(a|s)\, L(s,a)da. \quad (14)$$

Policy gradient techniques are utilized in our paper, in which the agent, which adjusts the parametric policy π_ψ , ψ is a group of parameters. The agent adjusts the policy using the gradient descent on the loss J due to ψ, which involves approximating the gradient of $J$with respect to ψ:

$$\nabla_\psi J(s,\pi_\psi)= \int_{a\in A}^{\cdot} L(s,a)\nabla_\psi \pi_\psi(a|s)\, da$$

$$= \int_{a\in A}^{\cdot} \pi_\psi(a|s)L(s,a)\nabla_\psi \log(\pi_\psi(a|s))da \quad (15)$$

$$=\mathbb{E}_{\pi_\psi}\left[L(s,a)\nabla_\psi \ \log\left(\pi_\psi(a|s)\middle|s\right)\right]$$

Where the second equality tracks from $\nabla\ \log(u) = \frac{\nabla u}{u}$.

V. NUMERICAL RESULTS & DISCUSSION

A. Numerical results

TABLE I
THE RANGE OF PARAMETERS USED FOR THE ANALYSIS

Parameter	value
Diffusion coefficient (D)	79.4 $(\mu m)^2/s$
The radius of the receiver (r_r)	10 μm
Distance (d)	{1~10} μm
r_0	r_r + {1~10} μm
N^{Tx}	{2~200,1000}molecules
Detection threshold (ε)	{0~N^{Tx}}

In this section, we train the DRL autoencoder model illustrated above, and we discuss the error probability simply as BER. Moreover, we evaluate the BER performance over a range of SNRs and compare the performance of the MC system calculated from (7) with those derived from the DRL autoencoder.

We used Matlab for the MC system performance that uses the BCSK modulation, and a simple threshold demodulation technique and these results are stated as the theoretical result.

In the DRL, the autoencoder evaluation is performed over a Gaussian channel. We design T_x and R_x as the feed-forward DNNs that leverage only the FC layer. The K^{th} layer is described as

$$r_k = g(W_k r_{k-1} + b_k) \quad (16)$$

Where $W_k \in \mathbb{R}^{N_{k-1}} \times \mathbb{R}^{N_k}$ is the weight matrix, $b_k \in \mathbb{R}^{N_k}$ is the NN bias vector and $g{:}\, \mathbb{R} \to \mathbb{R}$ is the activation function. The trainable parameters of the fully connected layer are $x_k = \{W_k, b_k\}$. The activation function is generally selected to be nonlinear that is essential to achieve DNNs which can estimate an extensive variety of functions.

The BER evolutions of the proposed technique during the initial five hundred training iterations are described in Fig.6, normally over two hundred seeds. After approximately four hundred training iterations, no significant recital difference is observed. Also, we noticed that the convergence of the training is not very fast.

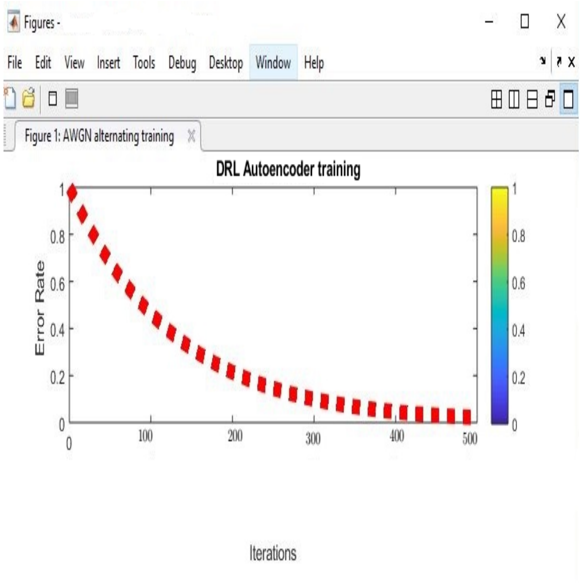
Figures -
File Edit View Insert Tools Debug Desktop Window Help
Figure 1: AWGN alternating training
DRL Autoencoder training
Error Rate
1
0.8
0.6
0.4
0.2
0
0
100
200
300
400
500
Iterations

FIGURE 6.BER duration initial five hundred training iterations systems.

In Fig.7 training by epoch using DRL (five hundred Iteration) and crosee entropy are shown.

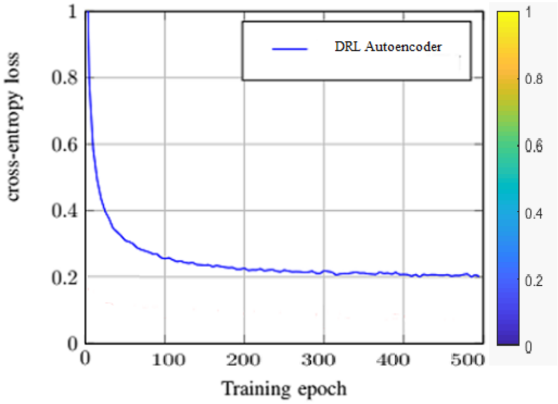

FIGURE 7. Average training loss entropy versus training epoch.

Fig.8 displays the SNR vs. BER outcomes for both the theoretically and DRL autoencoder. For the initial results, we achieved the slightly same performance for both methods at the low SNR and better performance from the DRL autoencoder over 7dB. We are confident with the extra hyperparameter tuning and longer training times at each SNR level; we can similarly achieve the same performance as an analytical technique for lower SNR values.

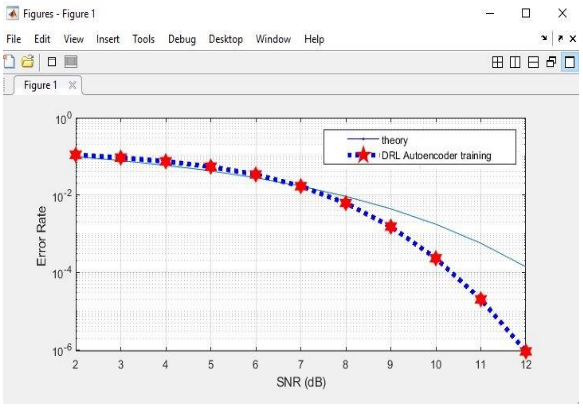

FIGURE 8. The BER of DRL autoencoder vs. the analytical method.

B. DISCUSSION

We successfully optimized MC as an end-to-end system without any knowledge about the channel model by using the DRL autoencoder that is capable of combating the problem of a missing channel gradient. In our case, the channel and the receiver are considered as the environment while the transmitter is considered as an agent. The traditional MC system depends on models and theories from channel modeling to information theory. These traditional models have serious issues, mainly in increasing the MC complexity. Recently, the field of machine learning has revealed that DRL can accomplish effective performance for several tasks, for example, automatic speech recognition and image classification [37]. Besides, it can be applied in the real-world application such as games, manufacturing, and healthcare. Therefore, the benefit of our result is that using the DRL autoencoder has many advantages:

1. Using the DRL shows its ability to optimize the performance of MC systems as end-to-end by learning methods instead of using mathematical models.
2. The use of the DRL in the MC system makes the communication system faster for the reason that the DRL comprises numerous phases and some computing processes that work in parallel. These guarantee speed in computing, capacity and data rates.
3. The use of the DRL in an MC system accomplishes enhancements in the performance of the system because the DRL attempts to optimize the performance for the end- to- end system without knowledge of the MC channel that helps to detect the free space molecular signal.

So, a joint scheme of DRL and MC would achieve a useful framework with the the trade-off that optimizes BER performance or complexity.

VI. CONCLUSION

In this article, we used a system Intel® Xeon® Processor CPU E5-2678 v3 @2.50GHzx48, Os type 64-bit, Memory 62.8GiB for evaluating the efficiency of the DRL autoencoder to achieve joint optimization transceivers techniques to implement the MC systems. We designed the actual computational framework for the MC which relies on the ideal information of the complete channel model. Based on this, we have optimized the BCSK modulation and the threshold demodulation, to reduce the error rate (BER). By using known learning techniques and by training a DRL autoencoder, we have discovered that the proposed technique achieved the same performance as the theoretical result on low SNR and better performance on high SNR. In future works, it should be possible to use advanced reinforcement learning techniques with our introduced technique to extend a new level of intelligent and increase the speed of convergence.

REFERENCE

[1] T. Nakano, A. W. Eckford, and T. Haraguchi, Molecular Communication. Cambridge, U.K.: Cambridge Univ. Press, 2013.

[2] Q. H. Abbasi et al., ``Nano-communication for biomedical applications: A review on the state-of-the-art from physical layers to novel networking concepts,'' IEEE Access, vol. 4, pp. 3920-3935, 2016.

[3] I. F. Akyildiz, M. Pierobon, S. Balasubramaniam, and Y. Koucheryavy, ``The Internet of bio-nano things,'' IEEE Commun. Mag., vol. 53, no. 3, pp. 32-40, Mar. 2015.

[4] H. B. Y. A. E. C.-B. C. a. W. G. N. Farsad, " A comprehensive survey of recent advancements in molecular communication," IEEE Communications Surveys & Tutorials, vol. 18, no. 3, p. pp.1887–1919, 2016.

[5] Nakano, T., Moore, M. J., Wei, F., Vasilakos, A. V., & Shuai, J., "Molecular communication and networking: Opportunities and challenges.," IEEE transactions on nano bioscience, vol. 11, no. 2, pp. 135-148., 2012.

[6] J. Ziv and M. Zakai, "On functionals satisfying a data-processing theorem," IEEE Transactions on Information Theory, vol. 19, no. 3, pp. Pp. 275-283, 1973.

[7] M. &. A. I. F. Pierobon, "A physical end-to-end model for molecular communication in nanonetworks.," IEEE Journal on Selected Areas in Communications, 2010.

[8] O'Shea, T., & Hoydis, J., " An introduction to deep learning for the physical layer," IEEE Transactions on Cognitive Communications and Networking, vol. 4, no. 3, pp. 563-575, 2017.

[9] E. Nachmani, Y. Be'ery, and D. Burshtein, "Learning to decode linear codes using deep learning," in Communication, Control, and Computing (Allerton), 2016 54th Annual Allerton Conference on. IEEE, 2016.

[10] S. Cammerer, T. Gruber, J. Hoydis, and S. t. Brink, " Scaling deep learning-based decoding of polar codes via partitioning," arXiv preprint arXiv:1702.06901, 2017.

[11] S. C. J. H. a. S. t. B. T. Gruber, " On deep learning based channel decoding," in Information Sciences and Systems (CISS), 51st Annual Conference on. IEEE, 1-6, 2017.

[12] F. Liang, C. Shen, and F. Wu, "An iterative BP-CNN architecture for channel decoding," IEEE Journal of Selected Topics in Signal Processing, vol. 12, no. 1, pp. 144-159, 2018.

[13] H. Kim, Y. Jiang, R. Rana, S. Kannan, S. Oh, and P. Viswanath, "Communication algorithms via deep learning," in The International Zurich Seminar on Information and Communication (IZS 2018) Proceedings. ETH Zurich, 2018, pp. 48-50., 2018.

[14] N. Samuel, T. Diskin, and A. Wiesel, "Deep MIMO detection," arXiv preprint arXiv:1706.01151, 2017.

[15] Y.-S. Jeon, S.-N. Hong, and N. Lee, "Blind detection for MIMO systems with low-resolution ADCs using supervised learning," in Communications (ICC), 2017 IEEE International Conference on. IEEE, 2017, pp. 1-6.

[16] S.-N. Hong, Y.-S. Jeon, and N. Lee, "MIMO systems with low-resolution adcs: Linear coding approach," in Communications (ICC), 2017 IEEE International Conference on. IEEE, 2017, pp. 1-6.

[17] N. Farsad and A. Goldsmith, "Detection algorithms for communication systems using deep learning," arXiv preprint arXiv:1705.08044, 2017.

[18] Cammerer, S., Dörner, S., Hoydis, J., & ten Brink, S, "End-to-end Learning for Physical Layer Communications," in In The International Zurich Seminar on Information and Communication (IZS 2018) Proceedings (pp. 51-52). ETH Zurich, 2018.

[19] T. J. O'Shea, K. Karra, and T. C. Clancy, "Learning to communicate: Channel auto-encoders, domain-specific regularizers, and attention," in Signal Processing and Information Technology (ISSPIT), 2016 IEEE International Symposium on. IEEE, 2016, pp. 223-228.

[20] T. J. O'Shea, T. Erpek, and T. C. Clancy, arXiv preprint arXiv:1707.07980, 2017, Deep learning based MIMO communications.

[21] Yilmaz, H. B., Chae, C. B., Tepekule, B., & Pusane A. E, "Arrival modeling and error analysis for molecular communication via diffusion with drift," in Proceedings of the Second Annual International Conference on Nanoscale Computing and Communication (p. 26). ACM., 2015, September.

[22] Dörner, S., Cammerer, S., Hoydis, J., & ten Brink, S. (2018). Deep learning based communication over the air. IEEE Journal of Selected Topics in Signal Processing, 12(1), 132-143.

[23] O'Shea, T. J., Erpek, T., & Clancy, T. C, "Physical layer deep learning of encodings for the MIMO fading channel," Communication, Control, and Computing (Allerton), 2017 55th Annual Allerton Conference on (pp. 76-80). IEEE., 2017.

[24] Farsad, N., Rao, M., & Goldsmith, A., "Deep Learning for Joint Source-Channel Coding of Text," arXiv preprint arXiv:1802.06832, 2018.

[25] Ometov, A., Levina, A., Borisenko, P., Mostovoy, R., Orsino, A., & Andreev, S, " Mobile social networking under side-channel attacks: Practical security challenges," IEEE Access, 5, pp. 2591-2601, 2017.

[26] Goutay, M., Aoudia, F. A., & Hoydis, J, " Deep Reinforcement Learning Autoencoder with Noisy Feedback," arXiv preprint arXiv:1810.05419., 2018.

[27] de Vrieze, C., Barratt, S., Tsai, D., & Sahai, A, " Cooperative Multi-Agent Reinforcement Learning for Low-Level Wireless Communication," arXiv preprint arXiv:1801.04541., 2018.

[28] Ye, H., Li, G. Y., Juang, B. H. F., & Sivanesan, K. (2018). Channel agnostic end-to-end learning based

communication systems with conditional gan. arXiv preprint arXiv:1807.00447.

[29] Damrath, M., & Hoeher, P. A, " Low-complexity adaptive threshold detection for molecular communication," IEEE transactions on nano bioscience, 15(3), pp. 200-208, 2016.

[30] Farsad, N., & Goldsmith, A, " Sliding Bidirectional Recurrent Neural Networks for Sequence Detection in Communication Systems," arXiv preprint arXiv:1802.08154., 2018.

[31] Qian, X., & Di Renzo, M, "Receiver Design in Molecular Communications: An Approach Based on Artificial Neural Networks," in 15th International Symposium on Wireless Communication Systems (ISWCS) IEEE, 2018.

[32] Yilmaz, H. B., & Chae, C. B, "Arrival modeling for molecular communication via diffusion," Electronics Letters, vol. 50, no. 23, pp. 1667-1669, 2014.

[33] Amin Zare, Ali Jamshidi, Alireza Keshavarz-Haddad, "Receiver design for a pulse for pulse position modulation technique in diffusion-based molecular communication," in Knowledge-Based Engineering and Innovation (KBEI), 2017 IEEE 4th International Conference on, IEEE, 2017, pp. 0729–0733.

[34] Goodfellow, I., Bengio, Y., Courville, A., & Bengio, Y, Deep learning, Cambridge: MIT Press, 2016.

[35] de Vrieze, C., Barratt, S., Tsai, D., & Sahai, A, " Cooperative Multi-Agent Reinforcement Learning for Low-Level Wireless Communication," arXiv preprint arXiv:1801.04541, 2018.

[36] Sutton, R. S., Mc Allester, D. A., Singh, S. P., & Mansour, Y, " Policy gradient methods for reinforcement learning with function approximation," Advances in neural information processing systems, pp. 1057-1063, 2000.

[37] Y. LeCun, Y. Bengio, and G. Hinton, Deep learning," Nature, vol. 521, no. 7553, pp. 436-444, 2015.

Soha Mohamed is a Ph.D. candidate at Harbin Institute of Technology, Harbin, China. Soha has a B.S. degree, pure mathematics & computer science. Soha obtained her M.S. degree in computer science from Harbin Institute of Technology. Research Interests includes but are not limited to: Internet of Bio-Nano things, machine learning, molecular communication, WSN.

JIAN DONG received a Ph.D. degree in computer science and technology from Harbin Institute of Technology, in 2008. From 2012 to 2017, he was an associate professor with the school of computer science and technology, HIT. Since 2017, he has been a professor with the school of computer science and technology, HIT. His research interests include a fault-tolerant computer, parallel computation and cloud computing.

A.R JUNEJO received the B.S. degree in Electronics Engineering and the Master’s degrees in Information & Communication Technology from Institute of Information and Communication Technology IICT, University of Sindh UoS, Jamshoro, Pakistan, in 2016 respectively. He is currently pursuing a Ph.D. degree with the Harbin Institute of Technology HIT, Harbin China. He has published over five technical articles in scientific journals and conference proceedings. His current research interest’s Neural Networks, Nanotechnology, and Molecular communication include computer networks.

De Cheng Zuo received the Ph.D. degree in computer science and technology from the Harbin Institute of Technology, in 2000, where he was an Associate Professor with the School of Computer Science and Technology, from 2003 to 2007, and has been a Professor with the School of Computer Science and Technology, since 2007. His research interests include computer architecture, fault-tolerant computing, computer systems performance evaluation technology, wireless sensor networks, wearable computing, and mobile computing.